

Performance Analysis in the Context of Layered Middleware *(or... why middleware is considered evil!)*

joint work with

Prof. Thomas M. Stricker

Dr. Michela Taufer

Institute for Computer Systems
Swiss Institute of Technology
ETH Zürich

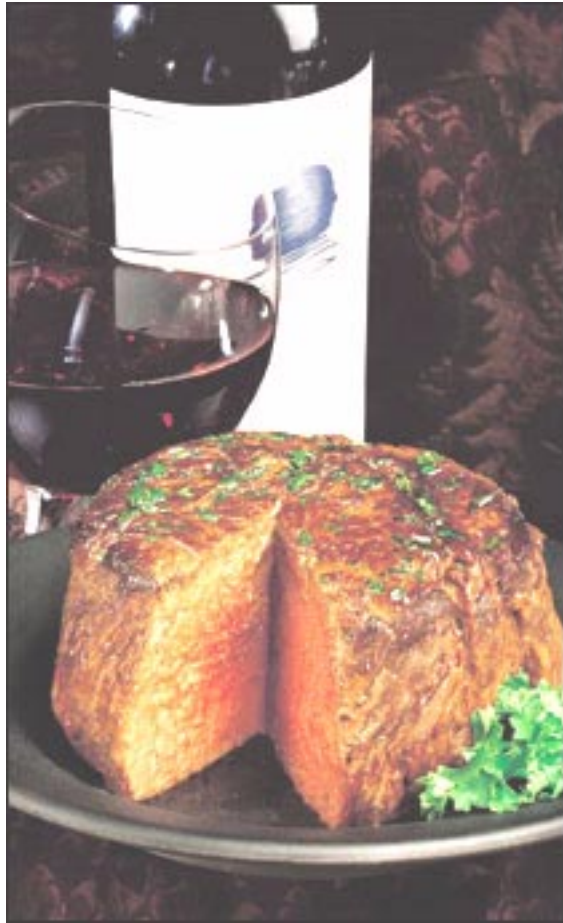
Scripps Institute & CSE Dept.
University of California
San Diego, CA

About our projects: <http://www.cs.inf.ethz.ch/CoPs>



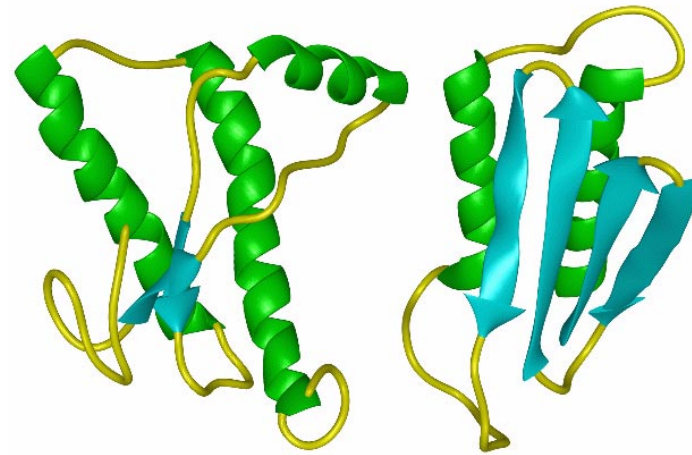
March 5, 2003
SOS7 Workshop

Molecular Biology Problem



Do you think it is safe to eat such a steak?

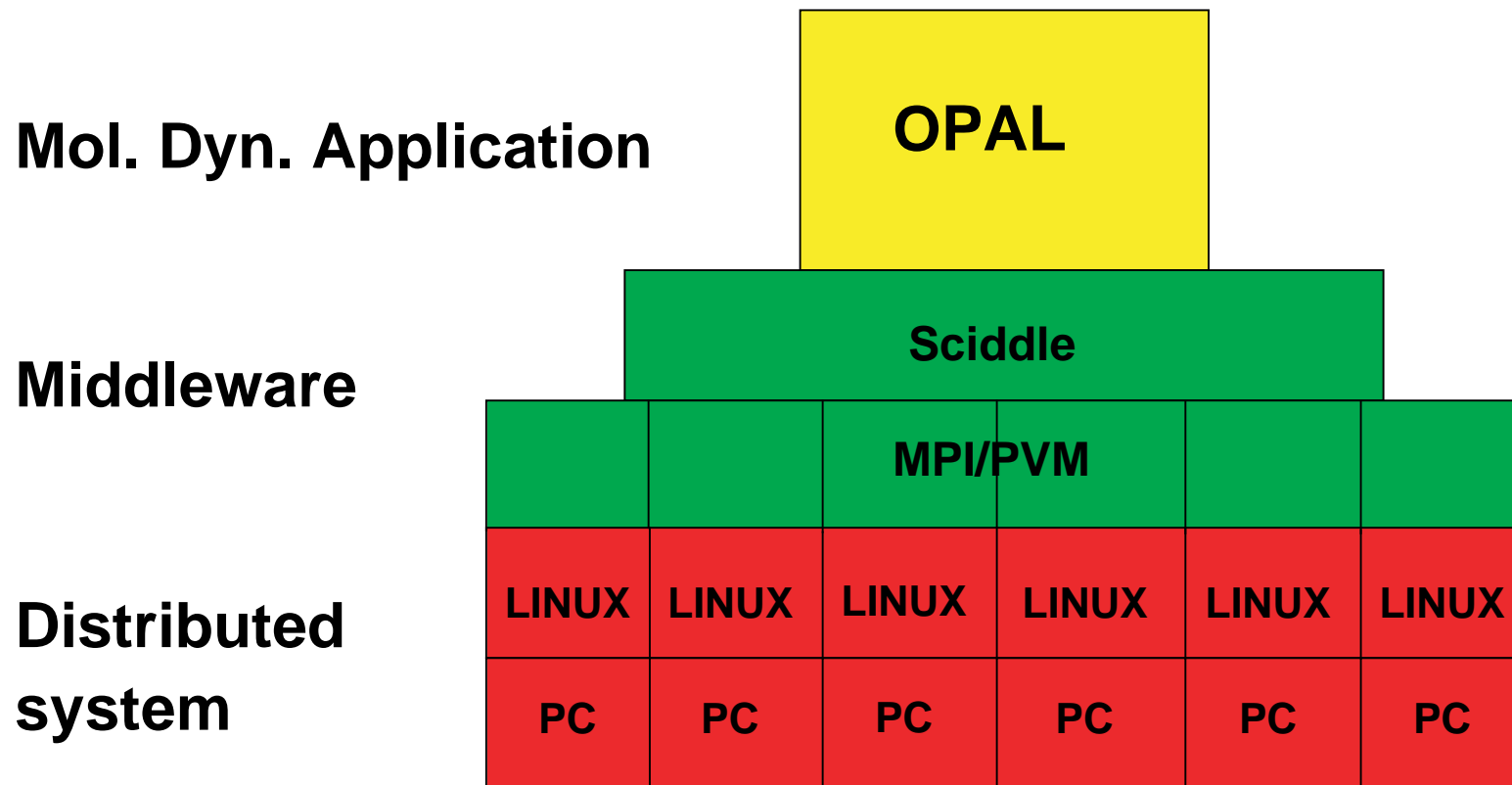
Have you ever heard about infectious proteins like prions?



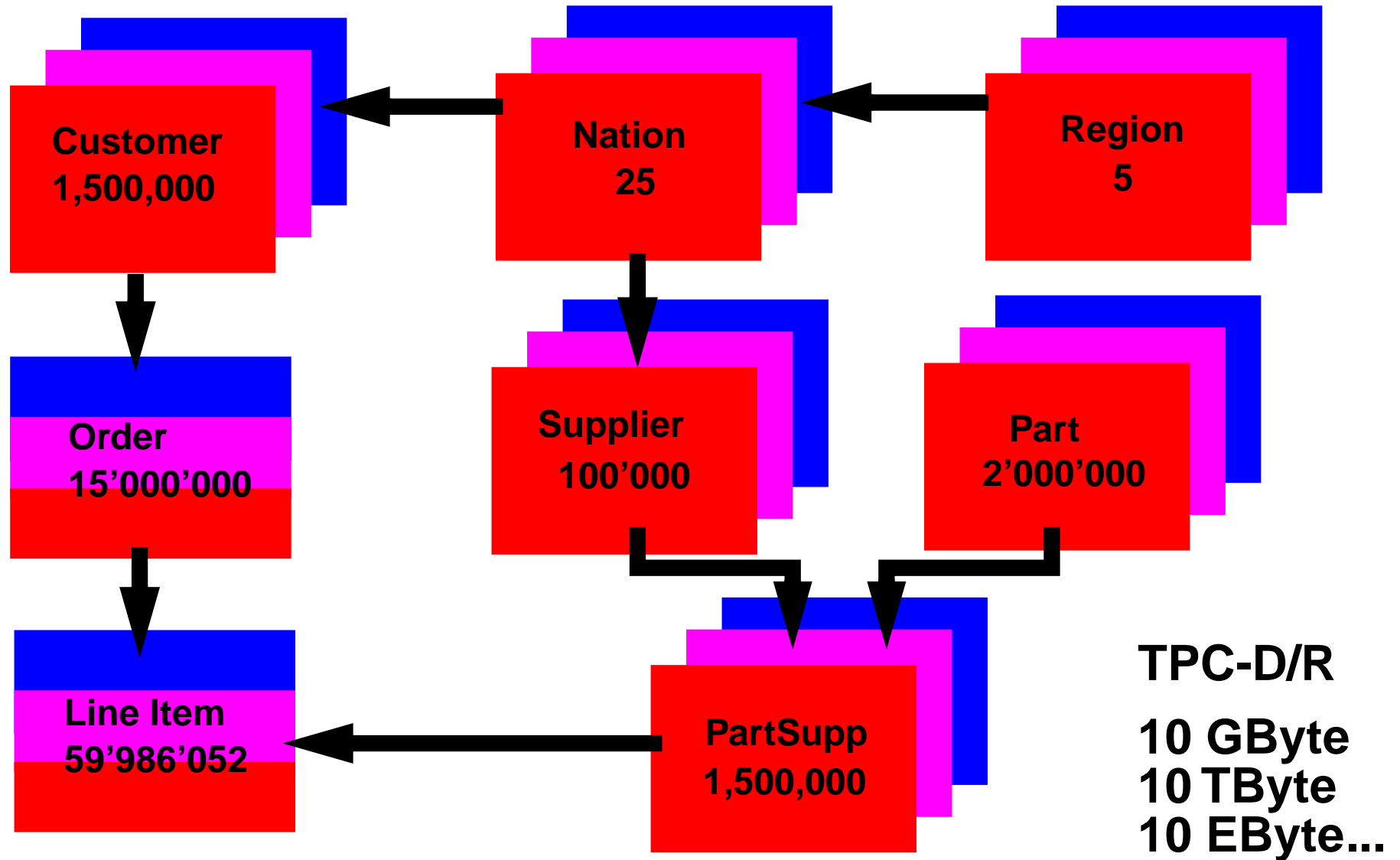
Two different foldings of the same prion...

- Which one is infectious? Why?
- What is the precise structure of the infectious ones?

Molecular Dynamics with Clusters of Commodity PCs



Data Mining (OLAP) Problem



Distributed DB on Cluster of Commodity PCs

OLAP Application

TPC-D

Middleware

TP-Lite

**Distributed
system**

ORACLE

LINUX

LINUX

LINUX

LINUX

LINUX

LINUX

PC

PC

PC

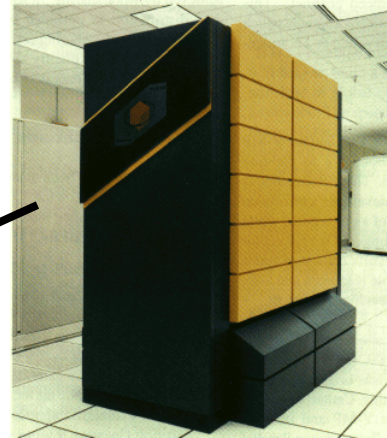
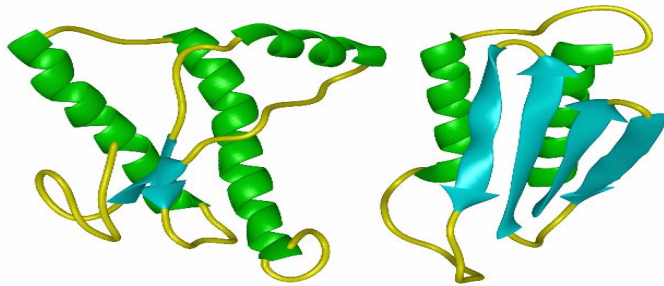
PC

PC

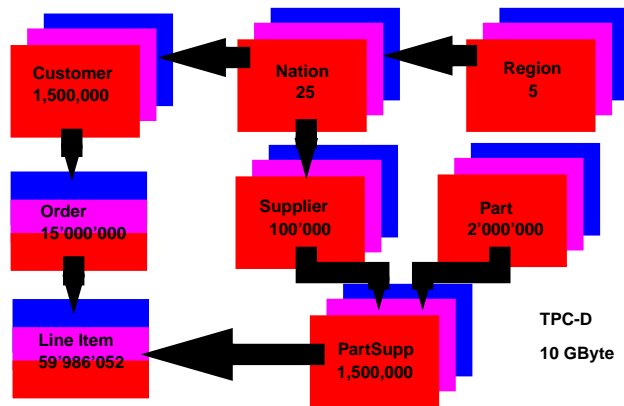
PC

Computer Architecture Problem

Distributed Comp. Code



Different Platforms

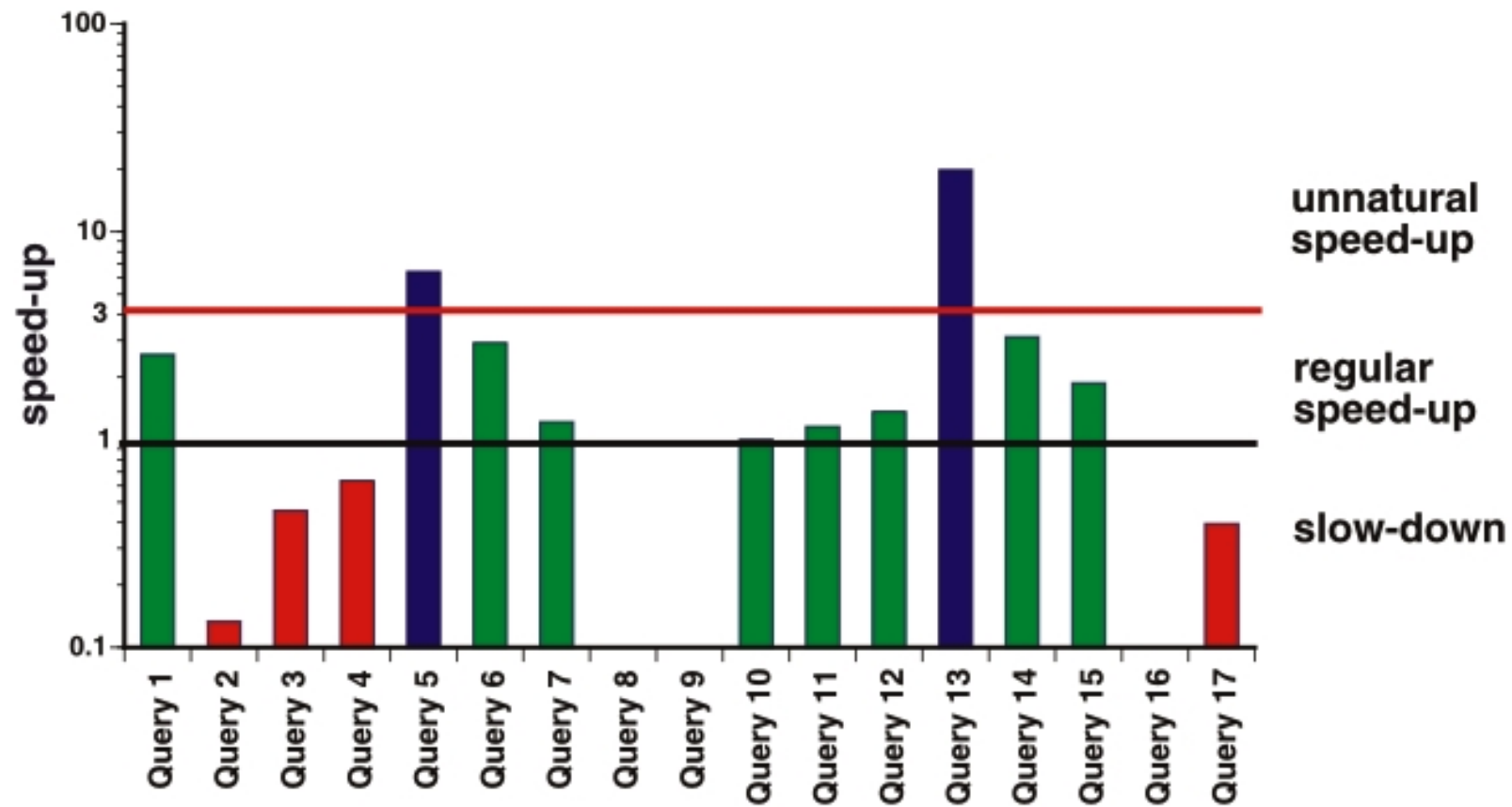


**Which ones
will work?**



**Which ones
will be most
cost effective?**

Results in Performance and Scalability



The Panel Questions?

What does work?

- **Distribution for better performance** - we can distribute and parallelize the computations.
- **Migrations from platform to platform** - we can migrate application codes from SMPs to Clusters of Commodity PCs and in some cases even onto Desktop Grids (*e.g. distributed.net*).
- **Benefits to Science** - the computations are highly useful.

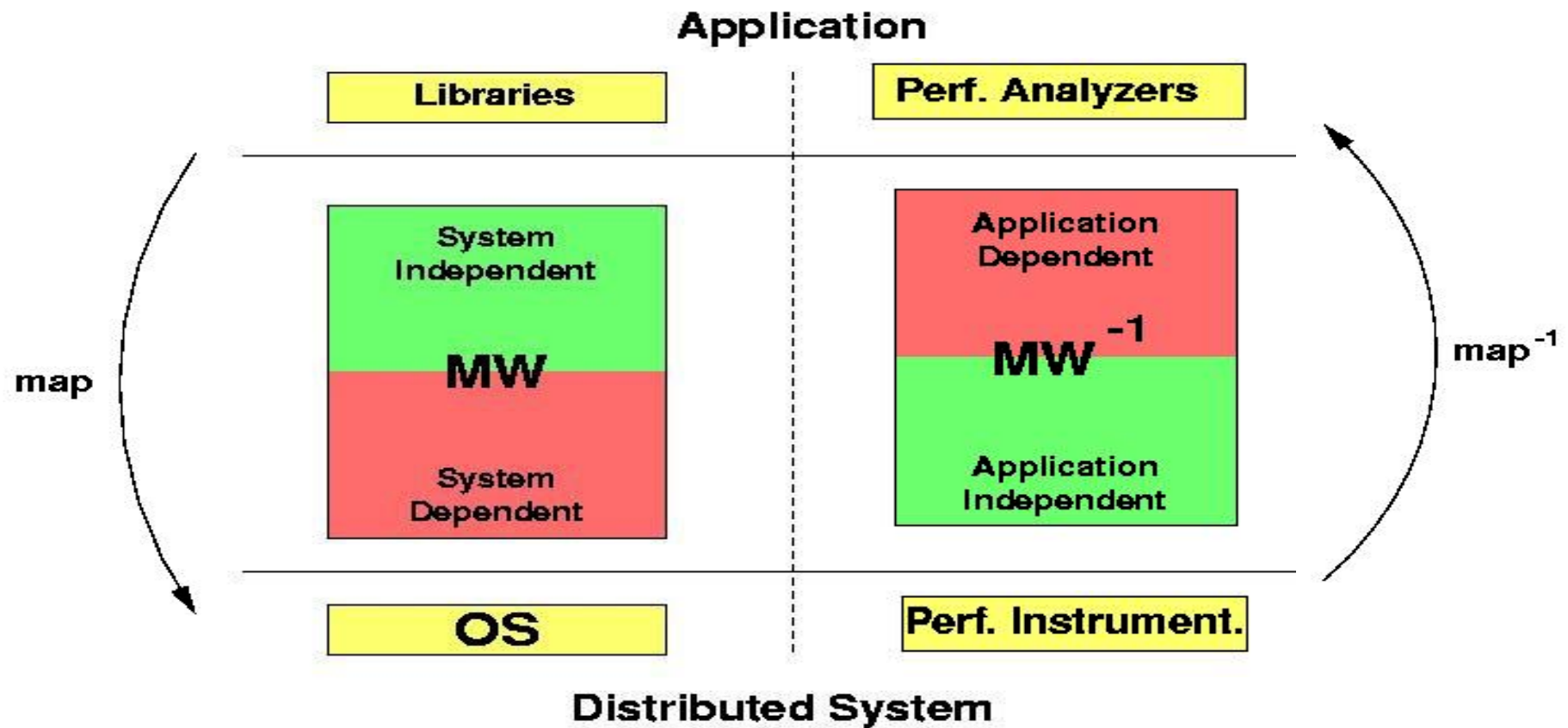
What does not work?

- **Predictable** and **analyzable** performance.

What lies ahead?

- **Fundamental work** on viability of a simple and effective performance models and analysis based on resource usage.

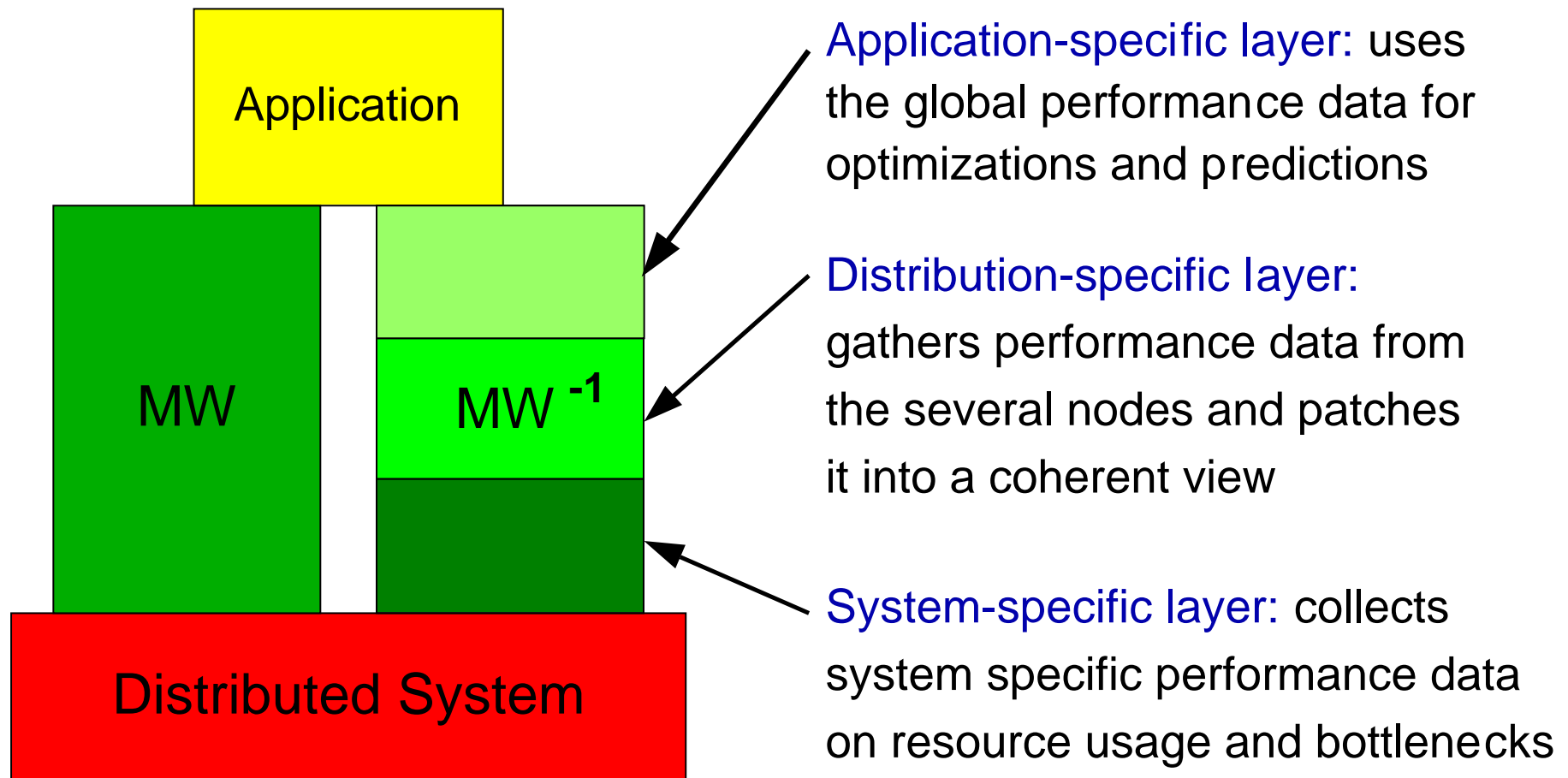
Approach: Complementing middleware (MW) with inverted middleware (MW^{-1})



Inverted Middleware

- The relation between middleware and inverted middleware is similar to the relation between compiler and debugger
- Inverted middleware comprises:
 - software instrumentation at the OS level,
 - tools for performance data collection and
 - analytical models

Internal Structure of Inverted Middleware



We actually did Inverted Middleware...

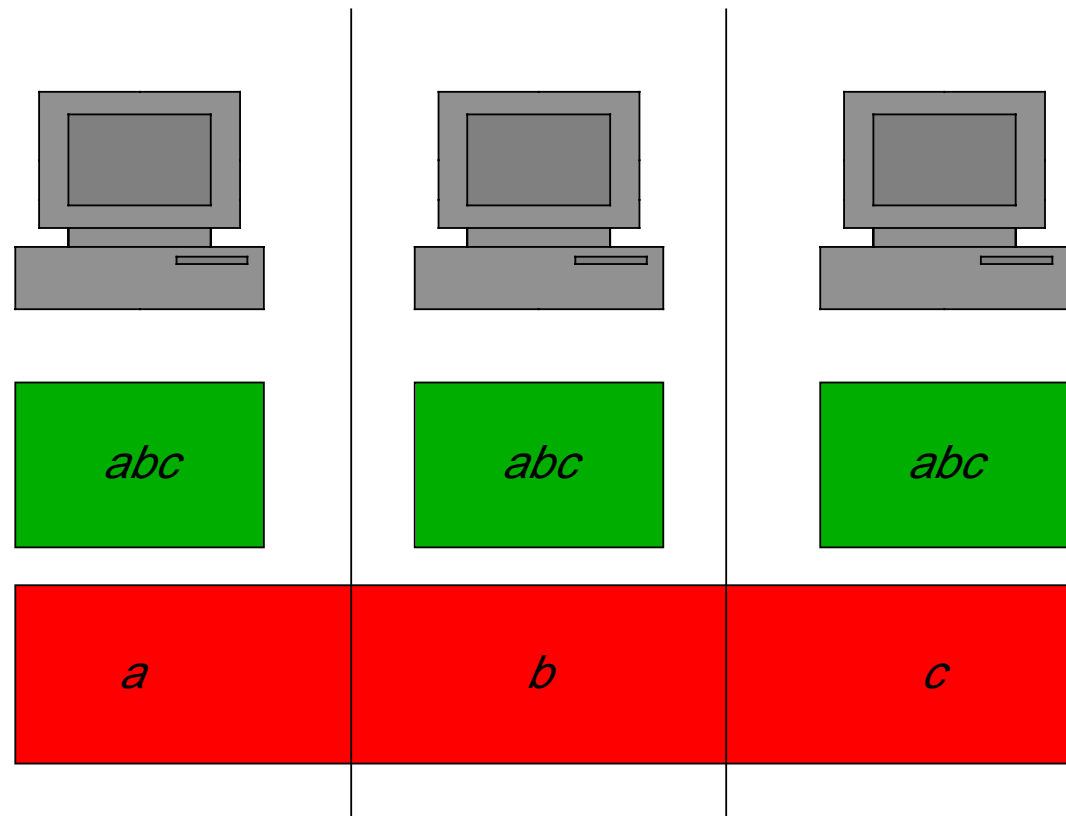
here is a sample performance evaluation (Oracle)

Database size: 10 GBytes

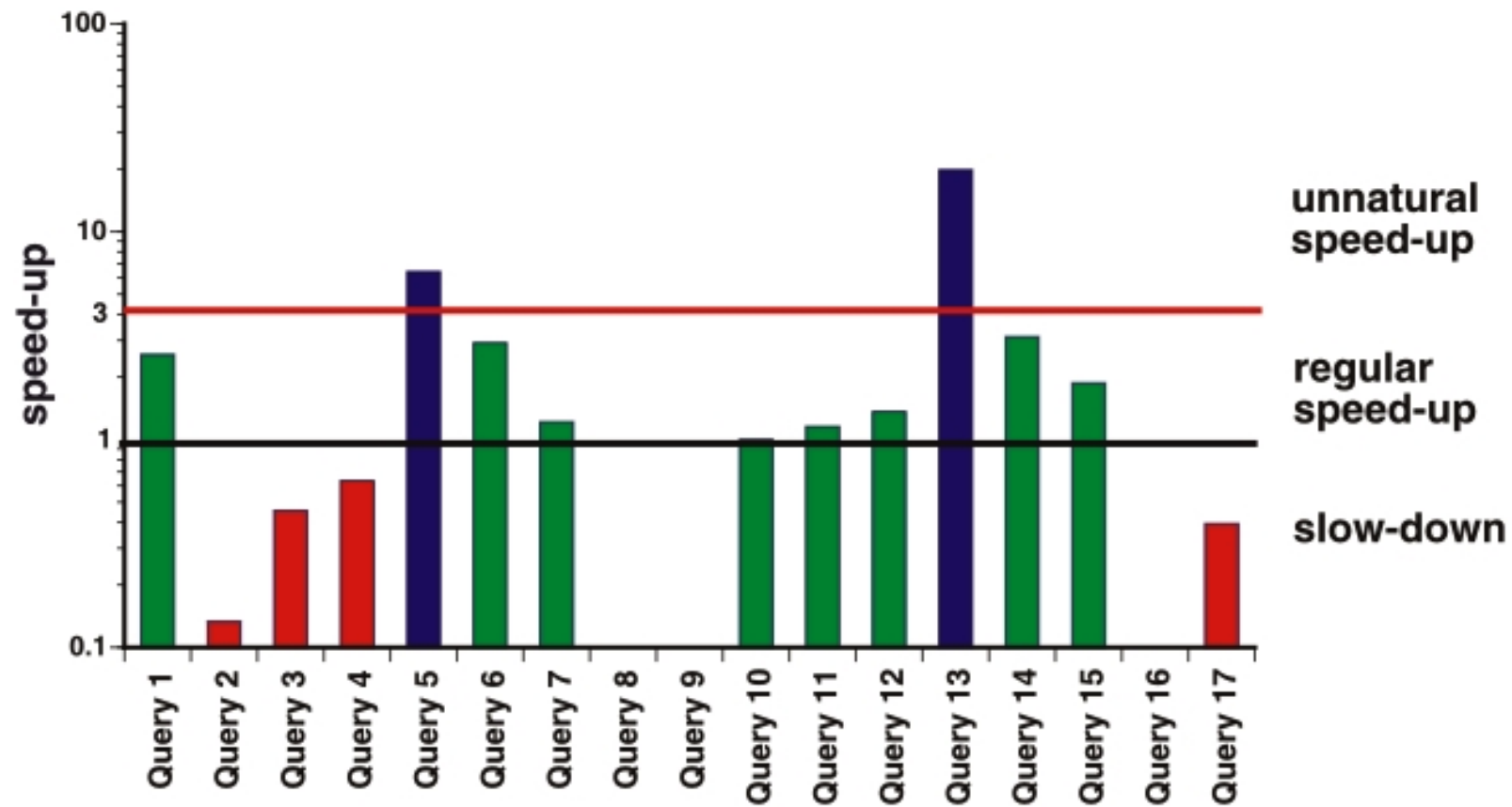
Cluster of commodity PCs

Small data tables (<1GBytes)
are fully replicated

Large data tables (>1GBytes)
are disjointly distributed



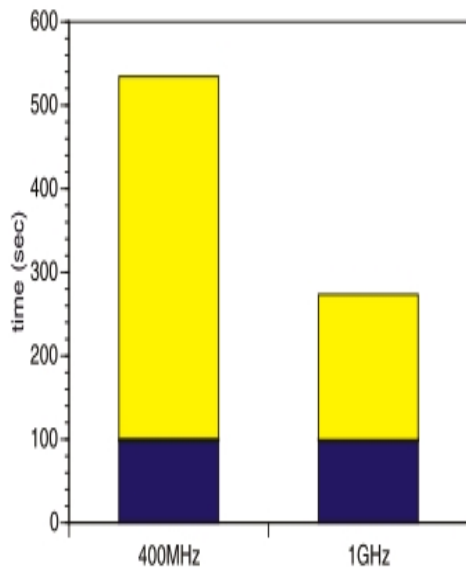
Results in Performance and Scalability



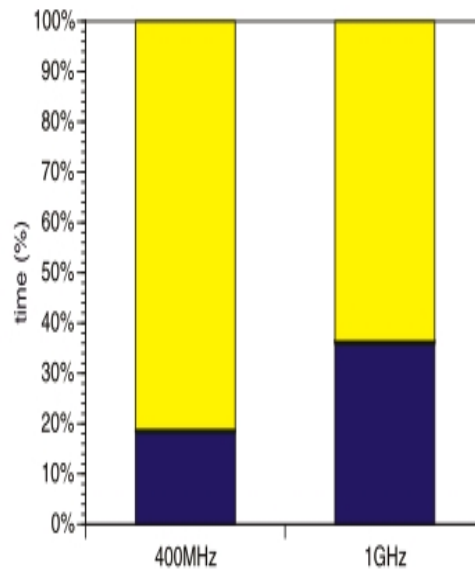
CPU-Limited Queries: **Clock-Rate** & **Scalability**

400MHz vs. 1GHz CPUs on 3 nodes

absolute (sec)

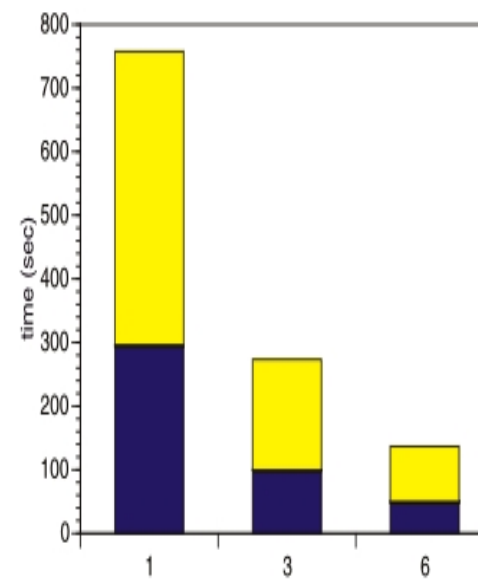


relative (%)

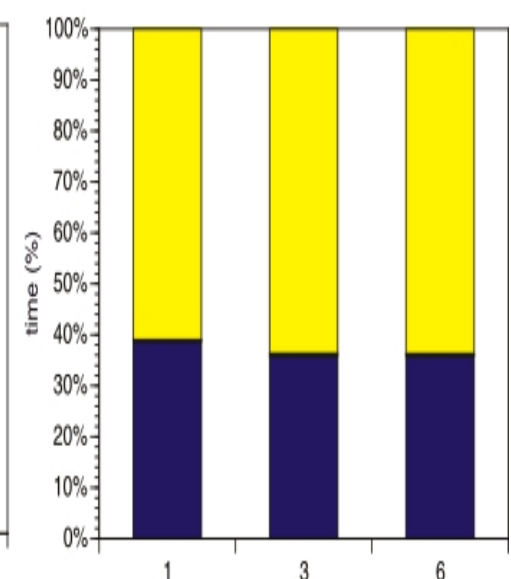


Scalability with 1GHz CPUs

absolute (sec)



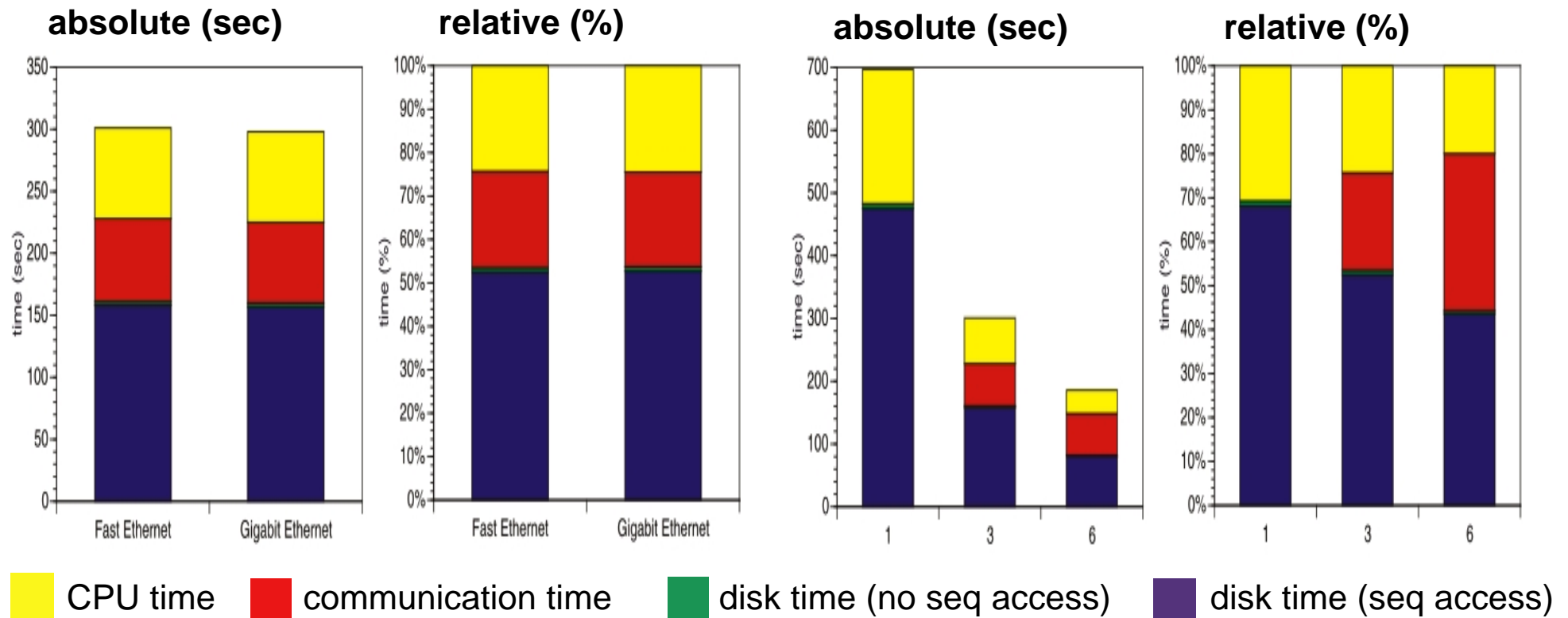
relative (%)



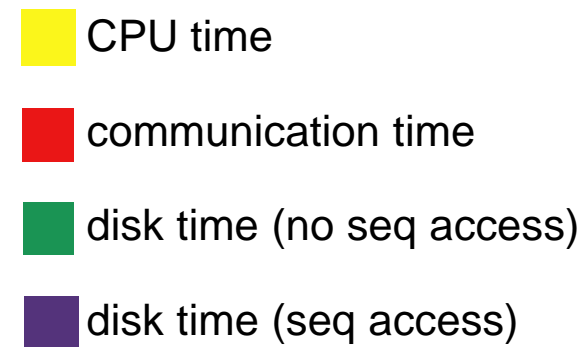
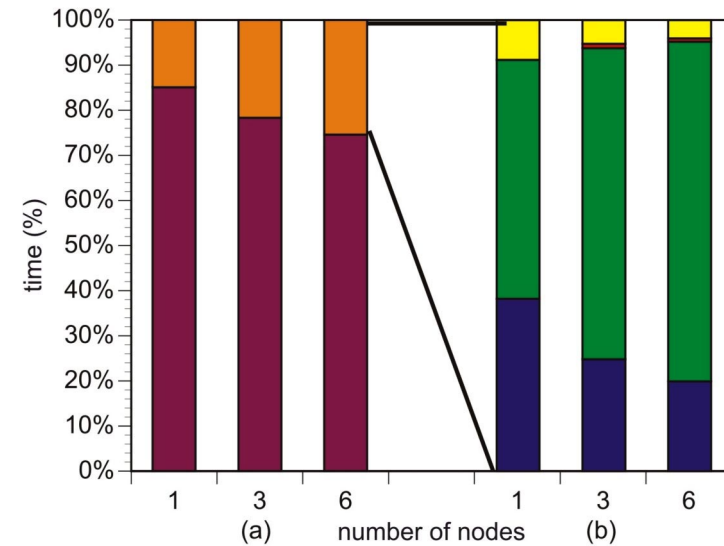
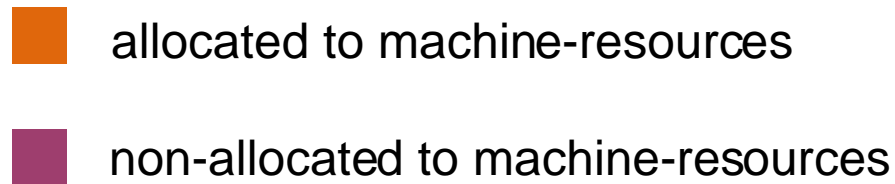
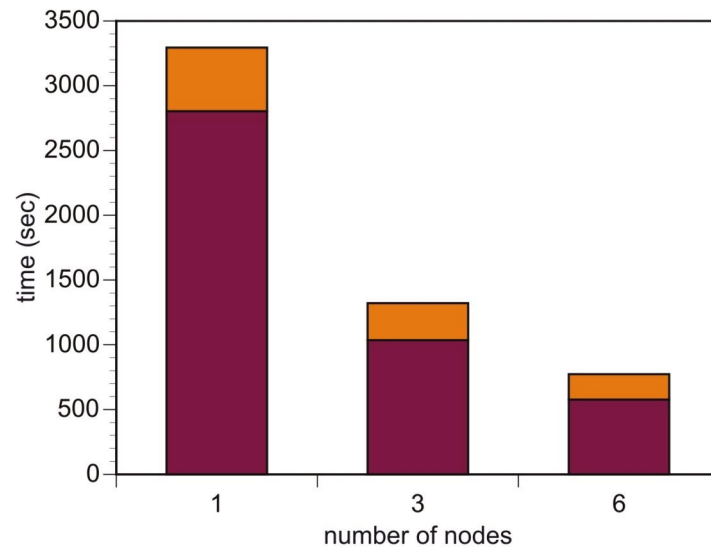
 CPU time  communication time  disk time (no seq access)  disk time (seq access)

Communication-Limited Queries: Resource Usage

Fast vs. Gigabit Ethernet on 3 nodes Scalability with Fast Ethernet



Middleware-Limited Queries: Resource Usage



Conclusions about Performance Impact of Middleware

- **Middleware** layered software system are normally decomposable in the software engineering sense (Holt/Courtois criterion).
- **Middleware** can be **inverted** with respect to performance impact - even when treated as a black box and when no source is available.
- **Our inverted middleware** proved to be a good tool and framework to analyze performance of distributed molecular biology codes and distributed databases on a variety of platforms ranging from **SMPs** to **Clusters** and **Desktop Grids**.

Many more **fundamental principles** and **new methods** need to be discovered. **Optimal performance** and **better efficiency** in middleware is far more than a question of hiring a few better “hackers”!